A Multipath Extension to the QUIC Module for ns-3

Shengjie Shu, Wenjun Yang, Jianping Pan, Lin Cai

Speaker: Shengjie (Shirley) Shu

June 29, 2023
QUIC Protocol

- A recently proposed transport protocol
- Address some limitations of TCP
  - connection establishment latency, head-of-line blocking, packet loss recovery, and mobility and handover support
- More promising for modern applications
- Standardized by IETF and integrated into HTTP/3
QUIC Protocol

- Key features

- **Connection Establishment:**
  - Zero round-trip time handshake
  - Encrypted connection with no additional handshake times
  - Transport Layer Security (TLS) 1.3 integration

- **Packet Header and Frame Structure:**
  - Payloads can contain various frame types
  - Stream multiplexing within a single connection

- **Loss Recovery and Error Control:**
  - Built-in retransmission and congestion control
  - Forward error correction (FEC) with an ACK frame
Multipath Scenarios

- End devices can connect with multiple network interfaces
- Potential benefits
  - Increase throughput
  - Uninterrupted communication and resilience
  - Load balancing
Multipath QUIC

- Extend the QUIC protocol to leverage multiple network interfaces simultaneously
- Aim to enhance performance, improve throughput, and fortify the protocol against link failures
- MPQUIC is under discussion by IETF
- Current experimental platforms for MPQUIC rely on either real systems or network emulators
- Absence of a hands-on MPQUIC simulation platform
Multipath QUIC

- MPQUIC follows the design logic of MPTCP and inherits the essential feature in QUIC
  - connection establishment, stream multiplexing, and frame structures

Figure 1: Structure of MPQUIC in comparison with others.
Challenge

- Address advertisement
- Path separation
- Algorithm extension
- Scheduler design
Challenge

- Address advertisement
- Path separation
- Algorithm extension
- Scheduler design

Maintain original QUIC transmission features
Our Implementation

- New classes
- New functions
- New variables

Figure 3: MPQUIC UML diagram (new classes, functions, and variables shown in italics).
Our Implementation

- New classes
- New functions
- New variables

Figure 3: MPQUIC UML diagram (new classes, functions, and variables shown in italics).
Our Implementation

- New classes
- New functions
- New variables

Figure 3: MPQUIC UML diagram (new classes, functions, and variables shown in italics).
Packet Header and New Frames

MPQUIC uses the frame structure to create additional sorts of frames for carrying multipath information.

Figure 2: MPQUIC Header and New Frames.
Path Identification

- \( m\text{\_pathId} \)

<table>
<thead>
<tr>
<th>Packet Header</th>
<th>Flag</th>
<th>Connection ID</th>
<th>Packet Number</th>
<th>Path ID</th>
<th>Payloads ...</th>
</tr>
</thead>
</table>

**MP_ACK**

<table>
<thead>
<tr>
<th>Frame Type</th>
<th>Path ID</th>
<th>Largest Acknowledged</th>
<th>Ack Delay</th>
<th>Ack Range</th>
<th>...</th>
</tr>
</thead>
</table>

**ADD_ADDRESS**

<table>
<thead>
<tr>
<th>Frame Type</th>
<th>Path ID</th>
<th>IP version</th>
<th>IP Address</th>
</tr>
</thead>
</table>

**REMOVE_ADDRESS**

<table>
<thead>
<tr>
<th>Frame Type</th>
<th>Path ID</th>
<th>IP version</th>
<th>IP Address</th>
</tr>
</thead>
</table>

Figure 2: MPQUIC Header and New Frames.
Path Management

- New class:
  - MpQuicPathManager
  - MpQuicSubflow

```python
m_enableMultipath=True
```

Figure 4: Procedures for subflow establishment.
Figure 5: State machine of a subflow.
Packet Scheduling

- MpQuicScheduler: m_schedulerType

<table>
<thead>
<tr>
<th>Scheduler</th>
<th>Round-Robin (RR)</th>
<th>Minimum-RTT (MRTT)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLEST</td>
<td></td>
<td>ECF</td>
</tr>
<tr>
<td>Peekaboo</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Congestion Control

- MpQuicCongestionOps
- QuicSocketBase::m_ccType
Implemented the fundamental transmission elements. Used in some internal research projects.

MPQUIC - 1.0
Fixed several issues.
Lightning talk in WNS3 2022.
Incorporated several scheduling.

MPQUIC - 1.2
Evaluated the implementation with more complicated scenarios.

Future work:
Align with subsequent IETF drafts.
Compare MPQUIC implementations in emulation scenarios.
Evaluation: Scalability

<table>
<thead>
<tr>
<th>Bandwidth</th>
<th>One-way Delay</th>
<th>Loss Rate</th>
<th>Data Size</th>
<th>Repeat</th>
</tr>
</thead>
<tbody>
<tr>
<td>5-5.5 Mbps</td>
<td>50-55 ms</td>
<td>0-0.08%</td>
<td>5 MB</td>
<td>50</td>
</tr>
</tbody>
</table>

Figure 7: Completion time and instantaneous throughput comparison for one, two, and four paths.
Evaluation: Congestion Control

<table>
<thead>
<tr>
<th>Bandwidth</th>
<th>One-way Delay</th>
<th>Loss Rate</th>
<th>Data Size</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>10-11 Mbps</td>
<td>10-11 ms</td>
<td>0-0.08%</td>
<td>Unlimited</td>
<td>50 s</td>
</tr>
</tbody>
</table>

Figure 8: Congestion window comparison for NewReno and OLIA
Evaluation: Schedulers

- Dominating Scenario

<table>
<thead>
<tr>
<th>Path</th>
<th>Bandwidth</th>
<th>One-way Delay</th>
<th>Loss Rate</th>
<th>Data Size</th>
<th>Repeat</th>
</tr>
</thead>
<tbody>
<tr>
<td>P0</td>
<td>5-5.5 Mbps</td>
<td>50-55 ms</td>
<td>0-0.08%</td>
<td>5 MB</td>
<td>50</td>
</tr>
<tr>
<td>P1</td>
<td>10-11 Mbps</td>
<td>10-11 ms</td>
<td>0-0.08%</td>
<td>5 MB</td>
<td>50</td>
</tr>
</tbody>
</table>

Figure 9: Completion time and instantaneous throughput comparison in the dominating scenario.
Evaluation: Schedulers

- Dominating Scenario

**Figure 11**: Received bytes of two paths in the dominating scenario

**Figure 12**: Received bytes of two paths under the dominating scenario with swapped setting after 5 seconds
Evaluation: Schedulers

- Competing Scenario

<table>
<thead>
<tr>
<th>Path</th>
<th>Bandwidth</th>
<th>One-way Delay</th>
<th>Loss Rate</th>
<th>Data Size</th>
<th>Repeat</th>
</tr>
</thead>
<tbody>
<tr>
<td>P0</td>
<td>5-5.5 Mbps</td>
<td>10-11 ms</td>
<td>0-0.01%</td>
<td>5 MB</td>
<td>50</td>
</tr>
<tr>
<td>P1</td>
<td>10-11 Mbps</td>
<td>50-55 ms</td>
<td>0-0.01%</td>
<td>5 MB</td>
<td>50</td>
</tr>
</tbody>
</table>

Figure 10: Completion time and instantaneous throughput comparison in the competing scenario.
Evaluation: Schedulers

- Competing Scenario

Figure 13: Received bytes of two paths in the competing scenario
Conclusions

❑ Provided a stable simulation platform of MPQUIC in ns-3
❑ Overcame the challenges of multipath transmission features
  ❑ address advertisement, path separation, and congestion control and scheduling algorithms
❑ Evaluated its correctness, scalability, and flexibility with a set of experimentations

Future work:
❑ Align with the future IETF draft
❑ Compare MPQUIC implementations in emulation scenarios
❑ Investigate better scheduling and congestion control techniques
Thank you!